KNN

- DADO UN DETERMINADO PUNTO KNN ENCUENTRA SUS K VECINOS MAS CERCANOS.

- BUSCA CADA VECINO DE MANERA LINEAL, ES DECIR POR FUERZA BRUTA Y ESTO PERJUDICA LA PERFORMANCE DEL ALGORITMO SI EL SET ES MUY GRANDE →

- PODEMOS USAR KNN PARA PROBLEMAS DE CLASIFICACIÓN Y REGRESIÓN.

- EL SET DE DATOS PUEDE SER DE M PUNTOS EN N DIMENSIONES.
  - SOLO ACEPTA FEATURES NUMERICOS Y NORMALIZADOS
  - EL ALGORITMO ES MUY SENSIBLE A FEATURES RUIDOSOS, POR LO CUAL ES MUY IMPO EL FEATURE IMPORTANCE.

- EL ALGORITMO NO ENTRENA, SOLO NECESITA DEL SET DE DATOS.

- 2 HIPER-PARAMETROS:
  - DISTANCIA A UTILIZAR
    - EUCLEDEANA
    - MANHATTAN
    - COSENO
    - JACCARD
    - HAMMIN
    - BM25
    - LEVEN

CLASIFICACIÓN: LE ASIGNAMOS AL PUNTO LA CLASE CON MAYORIA EN LOS K VECI.

REGRESIÓN: PREDECIMOS PARA EL PUNTO EL PROMEDIO DE LOS VALORES DE LOS K MAS CERCANOS.

- K: LA CANTIDAD DE VECINOS CERCANOS

  SI K ES MUY CHICO OCURRE OVERFITTING, SI K ES MUY GRANDE OCURRE UNDERFITTING.

EL M ÓPTIMO SE ENCUENTRA HACIENDO DISTINTAS PRUEBAS, UNA MUY ÚTIL ES LOOCV. ESTE CLASIFICA CADA PUNTO EN BASE A LOS DEMÁS.

- ESTOS 2 H-P TMB SE PUEDEN ENCONTRAR HACIENDO RANDOM SEARCH O GRID SEARCH.

- TENEMOS VARIAS OPTIMIZACIONES PARA MEJORAR EL PERFORMANCE CAUSADO POR LA BUSQUEDA LINEAL

  - KD-TREES
  - VP-TREES
  - LSH

  } INDICES ESPACIALES SON MUY EFICIENTES SI TENEMOS SETS 2D O 3D, MAYOR A ESTO SE DEGRADAN MUCHO Y EMPEORA LA PERFORMANCE.

↓

AYUDA A APROXIMAR A KNN, PERO HAY QUE TENER EN CUENTA SI KNN+LSH ES MEJOR QUE OTRO MODELO.

MALDICIÓN DE LA DIMENSIONALIDAD

" NO TODOS LOS ALGORITMOS FUNCIONAN BIEN EN CUALQUIER CANTIDAD DE DIMENSIONES"

↓

ESTA MALDICIÓN NO QUIERE DECIR QUE KNN FUNCIONA MAL EN MUCHAS DIMENSIONES, FUNCIONARIA MAL SI LOS DATOS FUERAN UNIFORMES, PERO NINGUN SET ES UNIFORME.
NO CONFUNDIR MALA PERFORMANCE CON LA MALDICIÓN!

- DADO UN SET DE DATOS DE M DATOS EN D DIMENSIONES ESTOS REPRESENTAN UN MANIFOLD DE K DIMENSIONES ($K \ll D$)
  ↓ VERDADERO GRADO DE LIBERTAD.